Mixed-Initiative Tool to Speed Up Content Creation in Physics-Based Games

César R. F. Gennaro Campos　Walter de Oliveira Sá　João M. Gusmão Teixeira　Levi H. S. Lelis

Departamento de Informática, Universidade Federal de Viçosa, Viçosa, Minas Gerais, Brazil

Abstract

In this paper we introduce a mixed-initiative tool to speed up the process of content generation in physics-based games. Our system allows the game designer to focus on the creative process of content creation by alleviating the burden of manually constructing game structures. We use a clone of Angry Birds called Science Birds as the object of our research. A user study shows the advantages of employing our mixed-initiative tool for creating Angry Birds levels. Namely, our study shows that people are able to create Angry Birds levels much more quickly with our mixed-initiative tool than with a baseline system we developed. Moreover, the levels created with our mixed-initiative tool are comparable in terms of quality with those created with the baseline system.

Keywords: Mixed-Initiative tool, procedural content generation, physics-based games.

1 Introduction

The creative process of computer game level design is often time consuming as the designer might have to perform multiple iterations between prototyping and playtesting before achieving the desired goals. The level design process can be specially time consuming in physics-based games such as Angry Birds (AB). This is because the elements of the game are subject to real-world effects such as gravity and friction, and it is often hard to predict the result of the interaction of several game elements without careful testing.

In this paper we present a mixed-initiative system to speed up the process of level design in physics-based games. Our goal is to allow the game designer to focus on the creative part of the process by having the computer doing most of the tedious and time consuming tasks for the designer. Although the object of our research is a clone of AB called Science Birds,[1] the system we introduce in this paper hinges on ideas that are likely applicable to other real-world problems that involve the collaboration between humans and algorithms, such as coordination of patient care in hospitals [1], autonomous driving [6], and general collaborative planning problems [4]. This is because we deal with general questions that appear in other human-algorithm interactions. For example, our system is concerned with balancing how much it is able to speed up the process of level creation while not interfering with the quality of the designer's work. Similar problems arise in automated driving, for example, where the system tries to perform most of the driver's tasks while not interfering with the passengers' safety.

The traditional process of creating AB levels consists of manually placing game elements on the screen so that they form stable structures (i.e., structures that do not move due to the effects of gravity). Our mixed-initiative tool partially replaces this process of manually placing elements on the screen by a drawing tool. The drawing tool allows the designer to quickly sketch the structures that will compose the level. The sketches are then automatically transformed into structures that use elements of the game. Once the basic structure of the level is created through the sketching tool, the designer is then able to perform fine adjustments to the structures thus created by manually adding and removing game elements. Our goal is to reduce the amount of work the designer has to perform in order to test an idea. We achieve that by reducing the number of elements the designer has to place on the screen. We believe our mixed-initiative tool might be able to enhance the designer's creativity by allowing them to quickly prototype different level ideas.

We performed a systematic user study to evaluate the mixed-initiative tool we introduce. Our primary goal with this study was to measure whether people are able to create levels more quickly with our mixed-initiative tool than with a baseline system that requires the manual placement of game elements. The results of our study show that our mixed-initiative tool is indeed able to significantly and substantially speed up the process of level creation. However, this speed up is meaningful only if the quality of the levels created with both approaches (mixed-initiative and baseline) are comparable. We derived a similarity metric for assessing how similar the levels created by the participants were from the levels they were told to create. The results suggest that people create levels with similar quality while using either system. The timing results taken together with the similarity results support the hypothesis that our system is able to speed up the level creation process while not interfering with the quality of the levels created.

2 Related Work

In this paper we consider a system as a mixed-initiative system if it makes decisions on its own while jointly solving a task, either with a human user or another system.

Tanagra is a mixed-initiative tool to design 2D platform levels [8]. Tanagra ensures that all levels created are playable and provides the ability for a human designer to look for many different levels that meet their design goals. Sketchbook Sentient is a computer-aided design tool for creating game maps that operates on high-level sketches instead of detailed maps [5]. A genetic-based tool presents maps suggestions in real time allowing the user to quickly replace parts of the map being created.

Ropossum is a tool to design levels for the physics-based game Cut The Rope [7]. Ropossum creates complete, playable Cut the Rope levels, and the designer is able to modify the levels created with an editing tool. Ropossum is perhaps the most similar system to the one we introduce in this paper. Nonetheless, our system substantially differs from Ropossum due to the number of game elements used in Cut the Rope and in AB. While levels of the former usually have a handful of elements, levels of the latter often have dozens of elements. The process of level prototyping is expected to be more time consuming in AB. Moreover, while Ropossum takes the initiative to fully create levels, the user of our system does not relinquish the design control to the computer, the user allows the computer to help with the level creation instead.

Butler et al. introduced a mixed-initiative system to generate levels for the educational game of Refraction [2]. Butler et al.'s system allows the designer to define a progression plan which is then satisfied by the computer.

[1] https://github.com/lucasnfe/Science-Birds

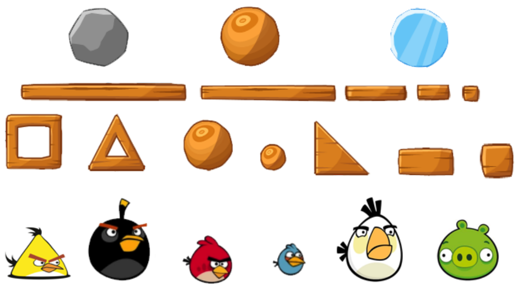


Figure 1: Elements used to build levels.

Figure 2: Example of a level being created with the manual module.

3 Angry Birds

Developed by Roxio, AB is a puzzle game based on physics. The player's objective is to destroy green pigs which can be guarded by block structures. The player is able to destroy blocks and pigs by throwing birds with a slingshot placed on the lefthand side of level. AB levels can be constructed by using the elements shown in Figure 1. Each element can be made of rock, wood, or ice, as demonstrated by rounded object in the first row of Figure 1. The second and third row of Figure 1 show all elements available to the designer in our mixed-initiative tool. Although all elements shown are made of wood, they are also available to be used as made of rock and ice. Our mixed-initiative tool is built on top of an AB's clone called Science Birds [3]. In addition to its material, each game element has properties such length l and width w. In our tool the objects can be rotated and we assume that the length l of an element denotes its longest side, while its width w its shortest side, independently of its position.

4 Manual Module for Content Creation

Our mixed-initiative tool for constructing AB levels is divided into two modules. The first module, which we call the manual module, allows the designer to manually construct AB levels. We describe the manual module in this section. The second module, which we call the mixed-initiative module, is described in Section 5. A user using the mixed-initiative features of our system uses both the manual module and the mixed-initiative module as the former allows the designer to edit the content generated by the latter. We also use the manual module as a baseline in our user study.

Figure 2 shows a screenshot of the manual module. The designer is able to select which element they will place on the level in the upper part of the screen. In this module the objects are placed manually by the designer who uses the computer's mouse to choose the element's location on the screen. On the top left corner of the screen there is a button in which the designer can choose the material of the elements (rock, wood, or ice). After placing an element on the screen the designer is able to rotate it. For example, the wooden blocks at the bottom of the structure were first placed in its default horizontal orientation, as depicted at the top of the figure, and then rotated vertically. At the bottom of the figure the designer can choose how many of each bird type the level will contain.

The designer is allowed to playtest the level being created by clicking on the button at the bottom-right corner of the screen. The playtest is important because the elements being placed in the level do not suffer the effects of friction and gravity. By playtesting the designer is able to verify whether the structures are stable. Should the designer discovers that a given structure is unstable, they can return to the module's editing mode to fix the source of instability. Another level property which is as important as stability is level feasibility, i.e., can the level be solved? The designer also ensures feasibility with playtesting. Finally, aiming at speeding up the process of level creation the manual module contains hotkeys for quickly selecting elements and changing the elements' materials.

5 Mixed-Initiative Module for Content Creation

In our mixed-initiative module the designer is able to quickly sketch AB structures with our drawing tool, and our system generates a structure according to the designer's sketch. An example of this process can be seen in Figure 3. The drawing tool is shown in Figure 3a, which initially displays an empty gridded canvas. The grid contains 35 $\times$ 45 cells. The designer is able to draw horizontal and vertical lines that can be edited with respect to their material—we intend to implement orientations other than vertical and horizontal for the lines drawn by the designer in future versions of our system. Purple lines represent elements made of rock, blue lines elements made of ice, and brown lines elements made of wood. Once the designer finishes sketching the lines representing the level's initial structure, the system generates the structure from the sketch. Figure 3b shows the structure generated from the lines shown in Figure 3a. In the following subsections we detail how our system creates the AB structures from the designer's sketch.

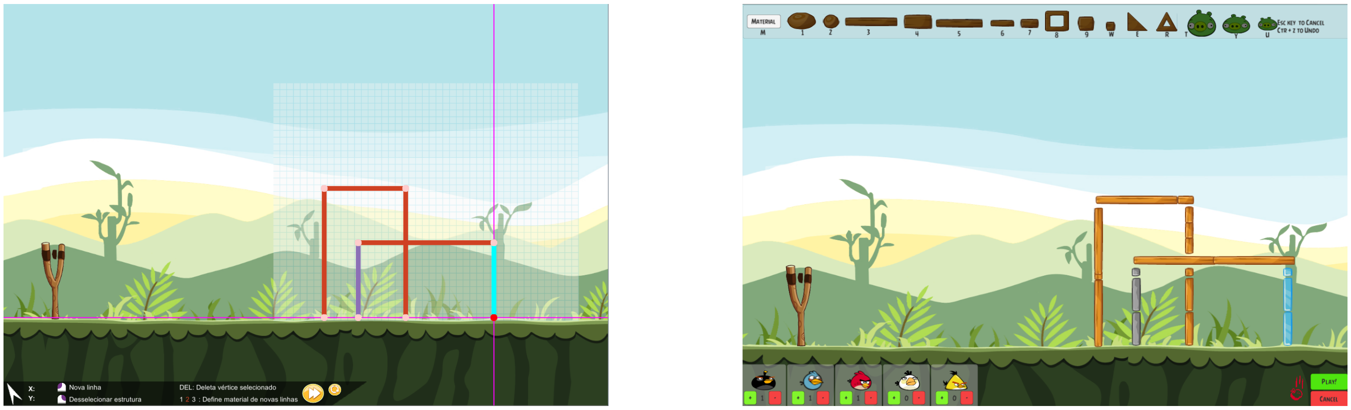
5.1 Sketch Representation

The sketching tool creates a collection L of line segments (henceforth line segments will be referred as lines) where each line in L is represented by a pair of points $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$. Each line i has a length c_i and the inclination d_i. The inclination will be either 0 or 90 degrees (horizontal or vertical), due to the constraint we impose in our drawing tool.

5.2 Handling Intersections

Our drawing tool allows the designer to create lines that intersect with each other; as an example, see the vertical and horizontal brown lines shown in Figure 3a. For every intersection of lines we alter the vertical line, by replacing it by two shorter lines, in a way that the intersection disappears. We replace the vertical instead of the horizontal lines because this way we are likely to reduce the amount of work the designer has to perform to make the resulting structure stable. Let us consider again the example of the lines in Figure 3a. Figure 3b shows the final result, where the vertical line was replaced and the horizontal line "cuts across" the original vertical line. If we did the opposite (i.e., replaced the horizontal line by two shorter lines), the designer would have to place supporters underneath the horizontal structures to make the whole structure stable. By contrast, since we replace the vertical line, the two shorter vertical lines already work as a support to the horizontal structure, thus likely reducing the designer's amount work.

Formally, if vertical line i defined by points $p_1 = (x, y_1)$ and $p_2 = (x, y_2)$ intersects a horizontal line at coordinate (x, y), assuming $y_1 < y_2$ (i.e., y_2 is above y_1) i is replaced in L by line i' defined by points $p'_1 = (x, y_1)$ and $p'_2 = (x, y-1)$ and by line i'' defined by points $p''_1 = (x, y+1)$ and $p''_2 = (x, y_2)$. The resulting collection L



(a) Drawing tool of the mixed-initiative module.

(b) Structure generated by our system from the drawing shown on the left.

Figure 3: Example of a structure created with the mixed-initiative module.

has no lines intersecting each other, and this is the collection used in the next step of our system, described below.

5.3 Brute-Force Search for Length Discovery

We use the elements shown on the second row of Figure 1 to create structures according to the lines in L provided by the drawing tool. The first element has dimensions 10.5×1, the second 8.5×1, the third 4.5×1, the fourth 2×1, and the last 1×1, here the first number represents l and the second w (we assume the elements to be placed horizontally, as shown in Figure 1). The dimension values of the elements are approximate as they depend on the bounding boxes used in the game. We discovered empirically that by using these values and removing one unit from the length of each line in L often result in visually pleasing structures. This reduction of one unit is performed for all lines in L.

Our mixed-initiative tool chooses a sequence of elements to form each line drawn by the user. For example, should the user draw a vertical line of length 15, our system will combine the smallest number of elements that together will form a line of length 15. An optimal solution is the combination of elements of length 10.5 and 4.5. Another solution is the combination of two elements of length 4.5 and three elements of length 2. We prefer the former combination because it uses only two elements, while the latter uses five elements. We minimize the number of elements composing the structures generated by our system because structures with fewer elements tend to be stronger and easier to be made stable. We call the problem of finding the set of AB elements that combined form a line of specific length l as the *AB length discovery problem*.

The main issue of minimizing the number of elements composing our structures is that the AB elements' length form a *non-canonical* problem. That is, similarly to the coin change problem, a greedy algorithm is unable to produce optimal solutions (i.e., solutions with the smallest number of elements). Since the lines drawn by the designer are defined by integer values, there will always be a solution to the AB length discovery problem. That is, the trivial solution of using a collection of 1×1 elements will always be applicable, independently of the line length.

In order to minimize the number of elements composing a given structure and still obtain answers in real-time, we devised a scheme in which we pre-compute the optimal combinations of elements for all sizes from 1 to 45 (recall that 45 is the longest line possible). The pre-computed combinations are stored in a lookup table which is used by the system while constructing structures from the designer's sketches. We use a simple depth-first search algorithm to enumerate all possible combinations of elements up to the maximal length of 45. Note that other schemes such as dynamic programming are possible, but since our problem is relatively small and the lookup table is computed only once, as a preprocessing step, we chose to use the simpler depth-first search approach. Our depth-first search approach finishes in approximately 3 minutes on a 2.7 GHz CPU and the resulting lookup table uses approximately 2 MB of memory. Once the lookup table is computed our system is able to efficiently access the combinations through the length of the desired structure—our lookup table is implemented as a hash table, which allows access to the element combinations in constant time.

5.3.1 Extending Horizontal Structures

In order to reduce the amount of work required to make the horizontal elements stable, we implemented the following enhancement in our algorithm. Whenever creating a structure representing a horizontal line i of length c_i, we create a structure with length equal to the structure with the smallest number of elements amongst the following lengths $\{c_i, c_i + 1, \cdots, c_i + \varepsilon\}$, where ε is a positive integer. For example, for horizontal line i with length $c_i = 16$ and $\varepsilon = 2$, we create a structure of size 17 instead of 16. This is because amongst the structures of lengths $\{16, 17, 18\}$, 17 is the length that requires the smallest number of elements to be formed—lines with lengths of 16 and 18 require three elements, while a line with 17 grid cells requires only two elements. The disadvantage of the ε-enhancement is that our system generates structures that might be slightly different from those sketched by the designer. The advantage is that the enhancement might reduce the number of elements composing the horizontal lines, which can reduce the amount of work the designer has to perform to ensure structure stability. We tested several ε-values in preliminary experiments and noticed that $\varepsilon = 3$ offered a good balance between not affecting the sketch created by the designer while still reducing the amount of work required to ensure structure stability.

6 User Study

Our user study follows a between-subject design, i.e., each participant evaluates either the mixed-initiative module or the manual module (baseline). We decided to use such a design to avoid carry-over effects. Since the mixed-initiative module also uses the manual module for the designer's final adjustments, the participants would likely construct AB levels faster when using the second system (either the baseline or the mixed-initiative system).

Each participant created three AB levels. For the first level the participants were free to create whichever structures and challenges they liked. The goal of this first level was to get the participant acquainted with the tool. Once the participant finished creating the practice level we showed on a separate screen the picture of an AB

Table 1: Scenario creation time in seconds.

	Scenario 1	Scenario 2
Mixed Initiative	205.55 ± 80.21	441.31 ± 140.72
Manual Module	290.34 ± 89.80	720.36 ± 409.28

level and asked the participant to reproduce that level as quickly and as accurate as possible; we call Scenario 1 (S1) this level. Once the participant finished reproducing S1, we showed the picture of another level to be reproduced, which we call Scenario 2 (S2). Scenarios 1 and 2 differ considerably in size: the former has 24 structural (wooden) elements, while the latter has 58. We measured the time it took for each participant to complete their reproduction of S1 and S2. After reproducing S1 and S2 the participants answered a demographics questionnaire. In total the experiment had 24 participants, 12 for each system. The participants who evaluated the baseline were 26 years old on average, 9 were male and 3 were female. The participants who evaluated the mixed-initiative system were 25.75 years old on average, 8 were male and 4 were female.

The average timing results as well as the standard deviations of our user study are presented in Table 1. The values are presented in seconds. Welch's t tests indicate that the differences between the baseline and the mixed-initiative systems are significant with $p < .05$ in both scenarios. The timing difference between the two approaches in S1 has a Cohen's d-value of 0.98, which indicates a large effect size. The effect size is smaller for S2, as the Cohen's d-value is 0.71, which indicates an effect size between medium and large. These results support our hypothesis that our mixed-initiative system is able to significantly ($p < .05$) and substantially (effect sizes in the medium and large marks) speed up the process of prototyping AB levels.

Table 2: Similarity results in Scenario 1

	Mixed Initiative	Manual Module
Structural Elements	4.08 ± 2.31	2.08 ± 3.34
Vertical Elements	21.25 ± 11.74	17.00 ± 38.87
Horizontal Elements	11.58 ± 13.63	11.50 ± 14.64
Pigs	0.17 ± 0.39	0.17 ± 14.64

Table 3: Similarity results in Scenario 2

	Mixed Initiative	Manual Module
Structural Elements	8.25 ± 4.94	7.00 ± 8.80
Vertical Elements	34.17 ± 32.90	44.18 ± 81.04
Horizontal Elements	76.75 ± 70.56	44.45 ± 56.36
Pigs	0.67 ± 0.78	0.18 ± 0.40

The significant and substantial positive timing results are meaningful only if the levels created by the participants are "similar enough" to the original levels. While we understand that the levels created by the participants do not have to match perfectly the levels they were told to reproduce, they have to have some structural similarity to allow for rapid tests of level prototypes.

We compute the similarity of the participants' levels and the original levels by computing the average absolute difference between the following level properties: total number of structural elements (wooden elements), total number of grid cells occupied by vertical elements, total number of grid cells occupied by horizontal elements, and total number of pigs. Intuitively, if a participant creates a level that is identical to the level they were told to recreate, then all absolute differences will be zero for that level. The average absolute differences for S1 and S2 are shown in Tables 2 and 3, respectively. As an example of how to read the table, a value of 4.08 for the mixed-initiative system in terms of structural elements means that the levels created by the participants either had 4.08 more structural elements than the original or had 4.08 fewer structural elements than the original on average.

The results shown in Table 2 suggest that the levels created by the participants using either system are equally similar to the original S1 level. Recall that S1 has 24 structural elements, which suggests that a difference of 2 elements (mixed-initiative absolute difference is 4.08 and the baseline absolute difference is 2.08 for structural elements) is not substantial. As can be observed in Table 3, the S2 levels created by the participants were less similar to the original than what was observed for S1. This is likely because S2 is more complex than S1. However, the results of both system are still comparable. The mixed-initiative system presented better results in terms of vertical elements, but worse results in terms of horizontal elements. The mixed-initiative participants had problems estimating the size of the horizontal lines in the level. The poor horizontal estimations led to narrower structures in which the pigs could not fit. As a result, the absolute difference of pigs was also larger for the mixed-initiative system. Despite these nuanced differences, the levels created by the participants using either system retained the main concepts of the original S2 level.

The similarity results taken together with the timing results support our hypothesis that our mixed-initiative tool might help to speed up the process of prototyping and testing game levels.

7 Conclusions

In this paper we presented a mixed-initiative system for rapid prototyping of levels of a physics-based game. The main goal with our system is to allow a quick test of the designer's ideas. Our mixed-initiative tool provides a way of quickly implementing different level ideas in domains where the process of level construction is typically tedious and time consuming. A detailed user study showed the advantages of our system over a baseline for level construction. Namely, our study showed that people using our mixed-initiative tool are able to create levels much more quickly than people using the baseline tool. Moreover, we showed that the levels created with the mixed-initiative tool are comparable in terms of quality with those created with the baseline.

References

[1] R. Bloss. Mobile hospital robots cure numerous logistic needs. *Industrial Robot*, 38(6):567–571, 2011.

[2] E. Butler, A. M. Smith, Y.-E. Liu, and Z. Popovic. A mixed-initiative tool for designing level progressions in games. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, pages 377–386. ACM, 2013.

[3] L. Ferreira and C. Toledo. A search-based approach for generating angry birds levels. In *Proceedings of the IEEE International Conference on Computational Intelligence in Games*, 2014.

[4] E. Kamar, Y. Gal, and B. J. Grosz. Incorporating helpful behavior into collaborative planning. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*, pages 875–882, 2009.

[5] A. Liapis, G. N. Yannakakis, and J. Togelius. Designer modeling for sentient sketchbook. In *2014 IEEE Conference on Computational Intelligence and Games*, pages 1–8. IEEE, 2014.

[6] B. K.-J. Mok, M. Johns, K. J. Lee, H. P. Ive, D. Miller, and W. Ju. Timing of unstructured transitions of control in automated driving. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 1167–1172. IEEE, 2015.

[7] N. Shaker, M. Shaker, and J. Togelius. Evolving playable content for cut the rope through a simulation-based approach. In *AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, pages 72–78, 2013.

[8] G. Smith, J. Whitehead, and M. Mateas. Tanagra: Reactive planning and constraint solving for mixed-initiative level design. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(3):201–215, 2011.